

REMARKS

The Applicant requests reconsideration of the rejection.

Claims 1-6 have been examined, and remain pending.

Claim 5 was deemed objectionable because of a typographical error in line 6.

Claim 5 has been amended to correct the error.

Claims 1-3, and 6 were rejected under 35 U.S.C. 103(a) as being unpatentable over Gregg, U.S. 5,938,786 (Gregg) in view of the Applicant's admitted prior art. The Applicant traverses as follows.

Each of the independent Claims 1-3 is directed to a data transfer method in which logical records are transferred between an initiator and a target. The logical records are batch-transferred in blocks, and the initiator confirms the transfer status at every batch transfer.

In rejecting Claims 1-3, the Examiner finds that Gregg discloses each feature of these claims except for posting the completion status of the transfer in a completion queue in the target when the logical record is received. However, the Examiner finds that the Applicant's admitted prior art (AAPA) discloses a completion queue for storing completion status before transferring the status back to the host.

The Applicant acknowledges that AAPA has known a target having a completion queue in which a completion status is stored for a queue pair existing on the target. However, AAPA does not disclose that the completion queue receives a completion status corresponding to a transfer request of individual logical records transferred batch-wise in a block. Rather, in AAPA, when all individual packets of a

logical record have been received correctly, the completion status is posted to the completion queue of the target, and a channel adaptor or the initiator is informed of the completed transfer.

According to AAPA, the initiator cannot start a transfer of the next logical record until confirming the notification of the fault-free transfer of the completion status of the entire logical record from the target, from each logical record transferred.

On the other hand, according to the present invention as set forth in the claims, the plurality of logical records are batch transferred in a block from an initiator to a target. The initiator confirms the transfer status of each batch transfer, and each logical record is transferred by a transfer request issued by the initiator. For each logical that meets a predetermined batch transfer condition, the target posts a completion status corresponding to the transfer request for the logical record to a completion queue existing in the target upon correct reception of a logical record (Claim 1). That is, logical records are batch-transferred, with each logical record in the batch being transferred by a transfer request. The target posts a completion status corresponding to each transfer request in its completion queue. The confirmation by the initiator is performed only at each batch transfer.

Gregg discloses a recovery scheme for damaged frames in a communication link. As noted in Column 4, lines 33-34 of the patent, Gregg discloses two types of frames: control frames, and information frames. Control frames do not have an information field, but comprise only a link-control word and a link-control CRC word.

An information frame, on the other hand, has a link-control word, a link-control CRC word, and an information field.

In the example of transferring multiple data areas (beginning on line 66 of Column 5), a write operation transfers two areas from an originator. In the transfer, a Message Command Block MCB and a first data area are sent. The first data area has an A-bit set to "1" indicating that more data areas are to follow. The recipient processes the first data area by moving it to main storage, freeing the buffer area for the receipt of the next data area. Next, the recipient sends an acknowledge frame ACK which contains no information field. The originator responds to the ACK by sending the next data area. The A-bit in this data frame is set to "0" if no more data areas are to follow.

Thus, Gregg is subject to the same problems as the prior art set forth in the Background section of the present specification. Gregg thus teaches away from the batch transfer of plural logical records in a block, wherein the initiator confirms the transfer status at every batch transfer, but each logical record is transferred by a transfer request, and for each logical record meeting a predetermined batch transfer condition, the target posts a completion status corresponding to the transfer request for the logical record to the target's completion queue upon correct reception of logical record. To emphasize the difference, Claim 1 (and Claims 2 and 3) has been amended to clarify a "logical record" and a "transfer request" in terms that more clearly distinguish Gregg.

Furthermore, in combination with AAPA, one of ordinary skill, at most, learns that a completion queue at Gregg's recipient receives a completion status of each data area, followed by the sending of the acknowledge frame ACK to the originator. Even in combination, there is no discussion of batch-wise transfer of logical records.

In this regard, the Applicant has noted the Examiner's comment that Gregg's "data field" shown in Fig. 4 are considered "logical records" transferred in a batch. Fig. 4 shows details of a link-control word 302, including a format that 402, type field 404, buffer set number field 406, A-bit 408, Start Bit 410, and Block Count field 412. Respectfully, the person of ordinary skill would never consider the combination of field comprising the link-control word 302 to be plural "logical records", or that the link-control word 302 could be a "batch" in a batch transfer of logical records.

As noted above, the link-control word is part of either a control frame or an information frame. The claims, however, define logical records as units of data transfer between the initiator and the target. The link-control word 302, on the other hand, has "data areas" that identify frame format and type, designate a buffer set area, and identify control states of the transceiver and link. Even if these fields were considered to comprise "units of data transfer", at most they would constitute part of a single logical record. Moreover, each of the fields, or data areas, is acknowledged by the recipient. There is no mention of acknowledgement at the end of a completion of an entire batch of so-called "logical records" comprising such data fields. Thus, Claim 1 is distinguishable from the combination of Gregg and AAPA.

Independent Claims 2 and 3 contain similar limitations. In addition, independent Claim 2 requires that, if the target detects a transfer error of the logical record in the middle of the batch transfer by the error check code, the target negates the reception of and stops posting the completion status of the logical record and subsequent logical records until the batch transfer terminates. Against this feature, the Examiner cites Gregg as teaching to send back an invalid buffer response INVRS such that, "if the command was for a batch transfer, e.g., the A-bit is asserted, this will [be] stopped by the quiescing of the original command and the data will be recovered via resending the data (Column 8, lines 60-65) after termination of the original request command."

However, Column 8, lines 60-65 states that the recovery action is to simply send the message again. On the other hand, Claim 2 requires that the target negate the reception of and stop posting the completion status of the logical record and subsequent logical records until the batch transfer terminates. Thus, this feature is not met by Gregg, whether taken individually or in combination with AAPA.

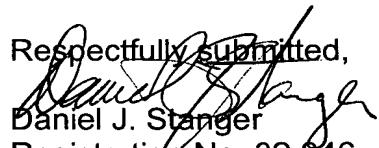
Claim 3 contains language distinguished above with regard to Claim 1 and Claim 2. In addition, Claim 3 requires that the target negate the reception of and stop posting the completion status of the logical record and subsequent logical records that are not permitted for reception by a value specified in a batch transfer condition field until the batch transfer terminates, if the target detects a transfer error of the logical record in the middle of the batch transfer. The Office Action does not address this feature of the invention, particularly the omission of a batch transfer

condition field. See, for example, batch transfer condition field 451 as part of payload 443, shown in Fig. 9 of the present application (it is noted that the batch transfer condition field need not be in the payload). In any event, the batch transfer condition field indicates whether the logical records are permitted for reception.

Dependent Claim 6 limits Claim 1 by permitting the initiator or the target to stop the batch transfer in the middle of a batch transfer by issuing a cancel request. In combination with the limitations of Claim 1, Claim 6 distinguishes Gregg and AAPA, which fail to disclose the batch transfer that can be canceled.

The Applicant acknowledges, with thanks, the allowability of the subject matter of Claims 4 and 5. These claims have been rewritten in independent form, including all limitations of original claim 1 (from which they were dependent), placing them in condition for allowance.

In view of the foregoing amendments and remarks, the Applicant respectfully requests reconsideration of the rejection and allowance of the claims.

Respectfully submitted,

Daniel J. Stanger
Registration No. 32,846

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.
1800 Diagonal Rd., Suite 370
Alexandria, Virginia 22313
(703) 684-1120
Date: December 20, 2005